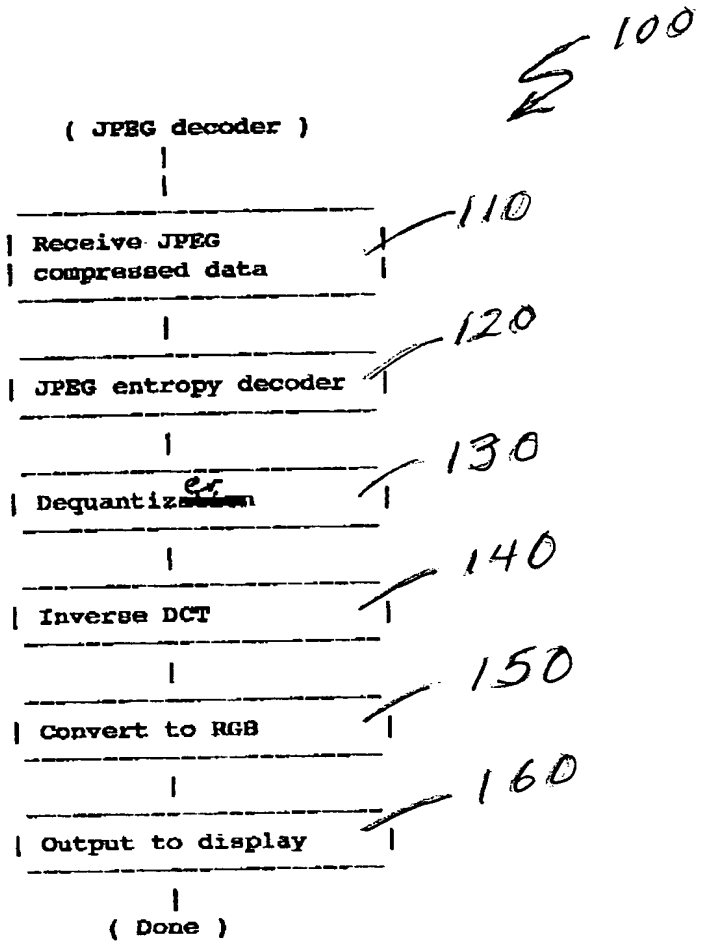
100
(JPEG decoder)
Receive JPEG compressed data
110
JPEG entropy decoder
120
Dequantization
130
Inverse DCT
140
Convert to RGB
150
Output to display
160
(Done)

FIG. 1

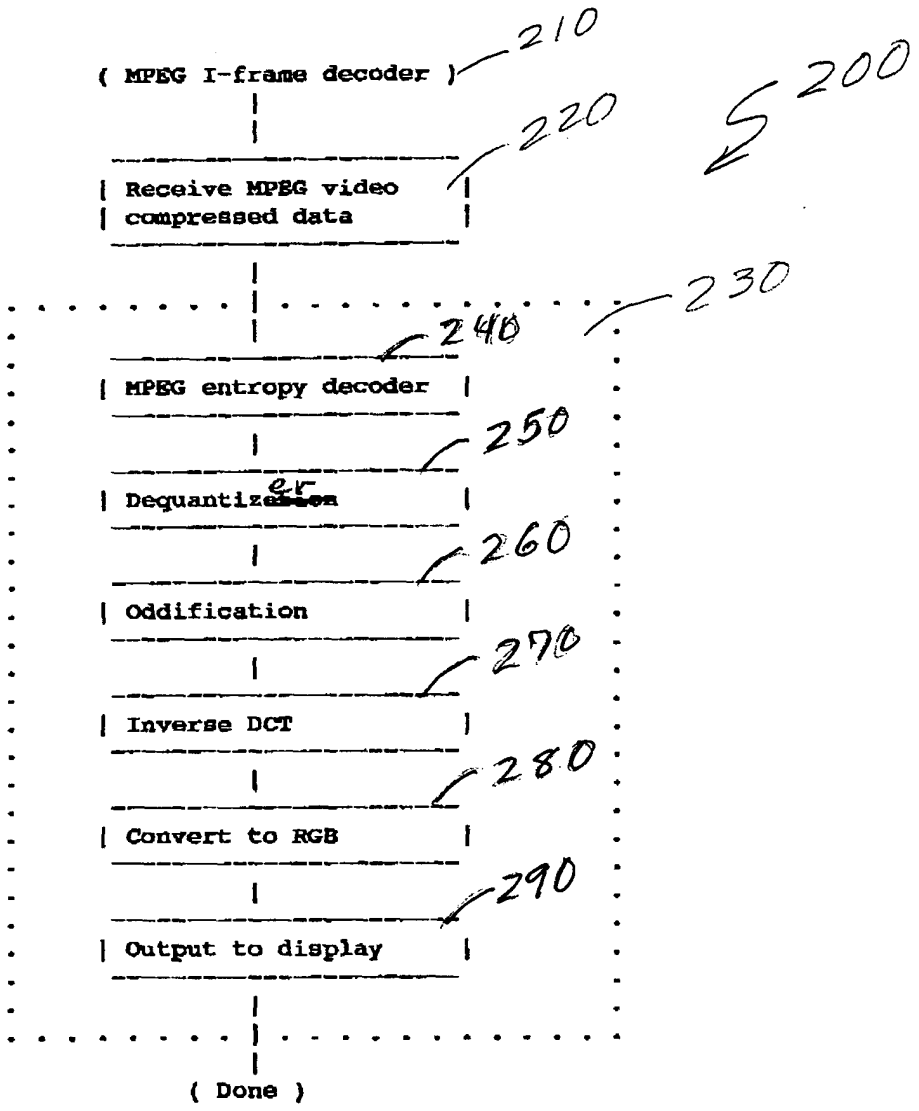
(MPEG I-frame decoder)
210
200
220
Receive MPEG video compressed data
230
240
MPEG entropy decoder
250
Dequantization
260
Oddification
270
Inverse DCT
280
Convert to RGB
290
Output to display
(Done)

FIG. 2

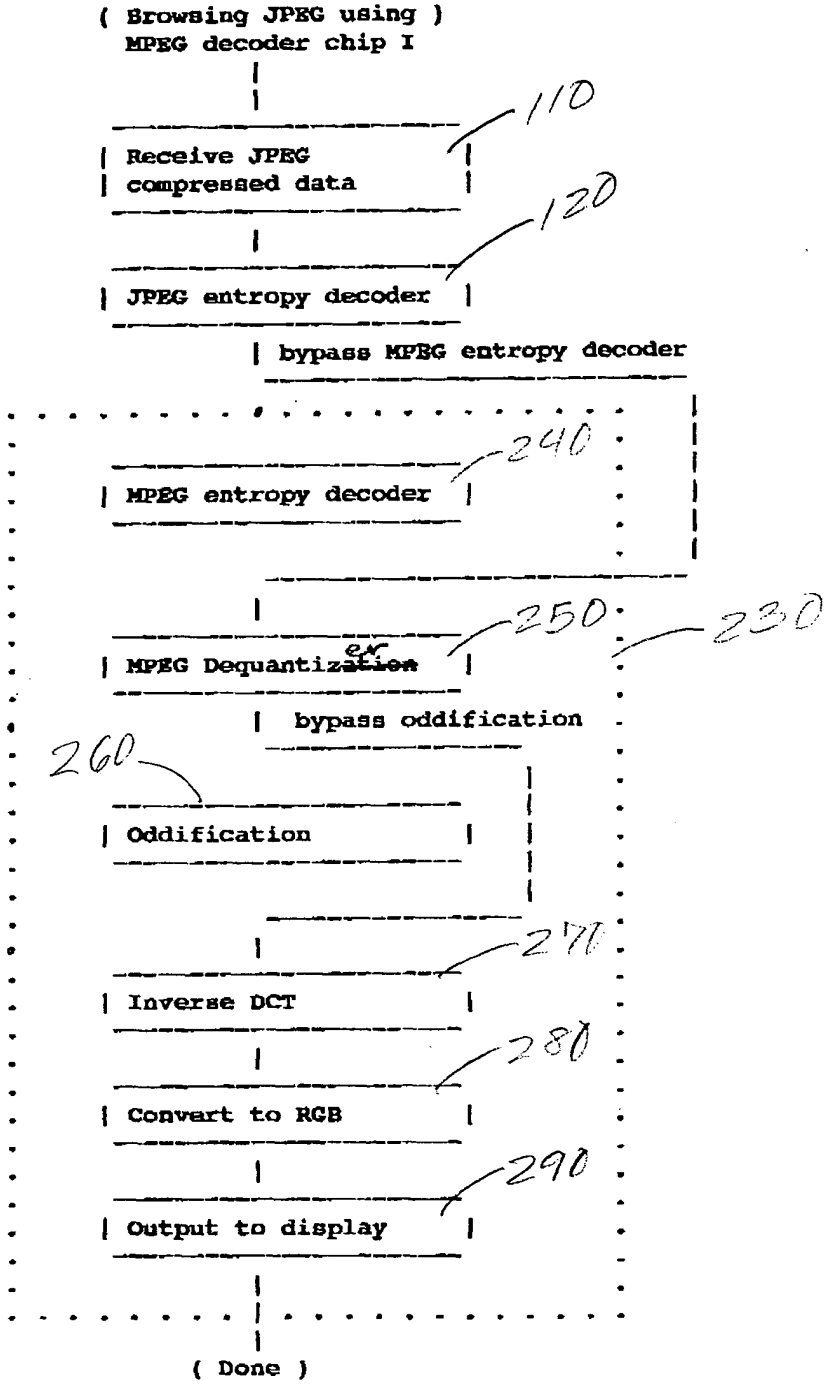
(Browsing JPEG using)
MPEG decoder chip I
110
Receive JPEG compressed data
120
JPEG entropy decoder
bypass MPEG entropy decoder
240
MPEG entropy decoder
250
230
MPEG Dequantization
bypass oddification
260
Oddification
270
Inverse DCT
280
Convert to RGB
290
Output to display
(Done)

FIG. 3A

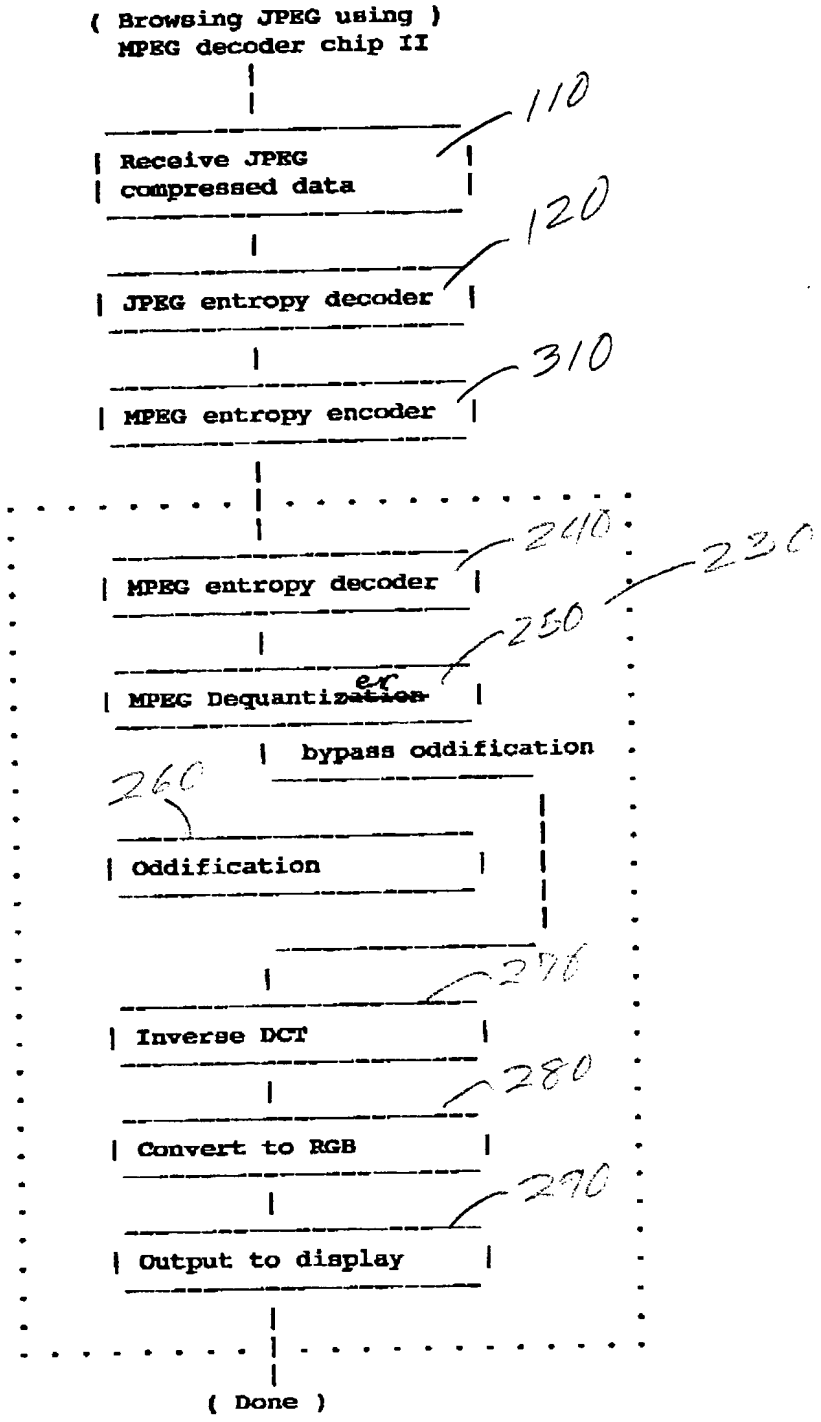
(Browsing JPEG using)
MPEG decoder chip II
110
Receive JPEG compressed data
120
JPEG entropy decoder
310
MPEG entropy encoder
240
MPEG entropy decoder
220
250
MPEG Dequantizer
bypass oddification
260
Oddification
270
Inverse DCT
280
Convert to RGB
290
Output to display
(Done)

FIG. 3B

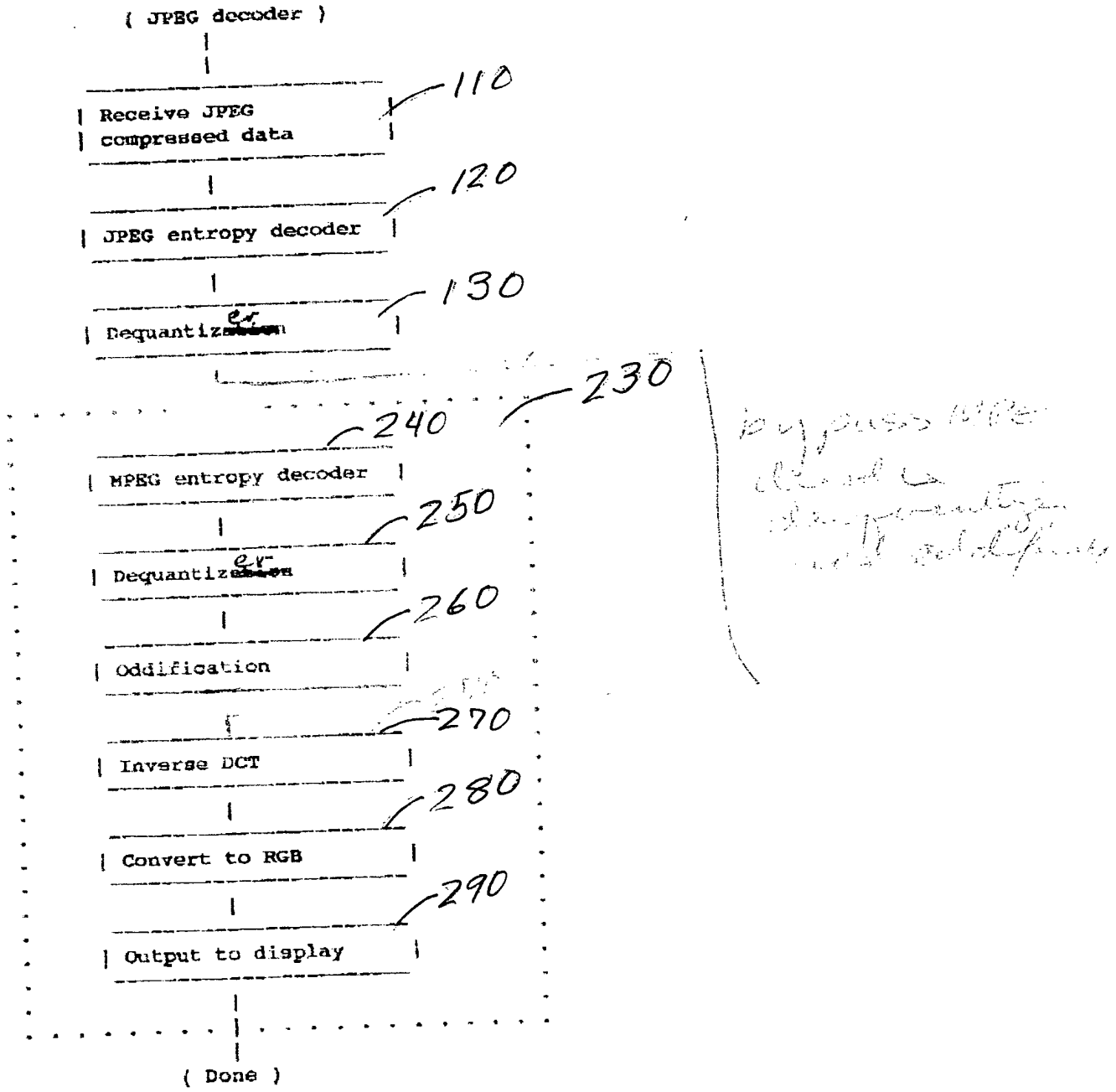
(JPEG decoder)
Receive JPEG compressed data
110
JPEG entropy decoder
120
Dequantizer
130
230
240
MPEG entropy decoder
250
Dequantizer
260
Oddification
270
Inverse DCT
280
Convert to RGB
290
Output to display
(Done)
bypass MPEG blocks

FIG. 3C

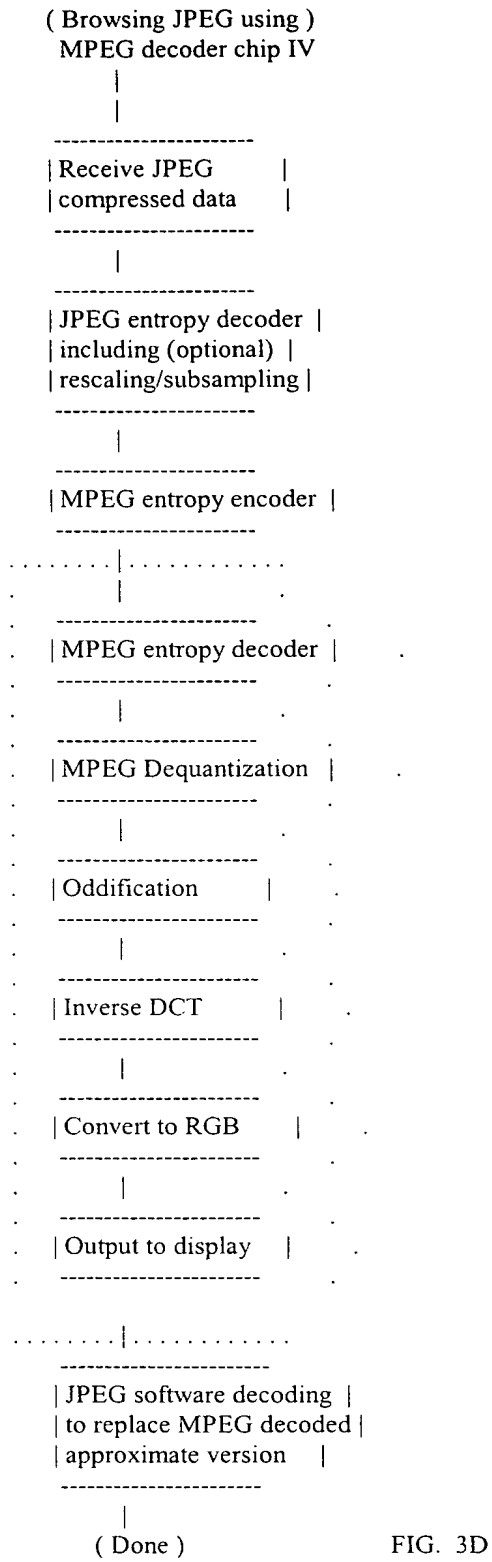
(Browsing JPEG using)
MPEG decoder chip IV
Receive JPEG compressed data
JPEG entropy decoder including (optional) rescaling/subsampling
MPEG entropy encoder
MPEG entropy decoder
MPEG Dequantization
Oddification
Inverse DCT
Convert to RGB
Output to display
JPEG software decoding to replace MPEG decoded approximate version
(Done)

FIG. 3D

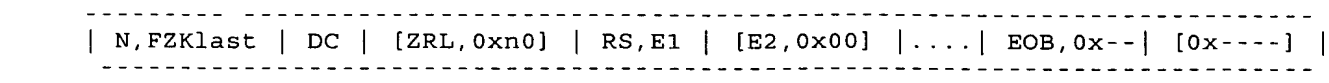

Figure 4

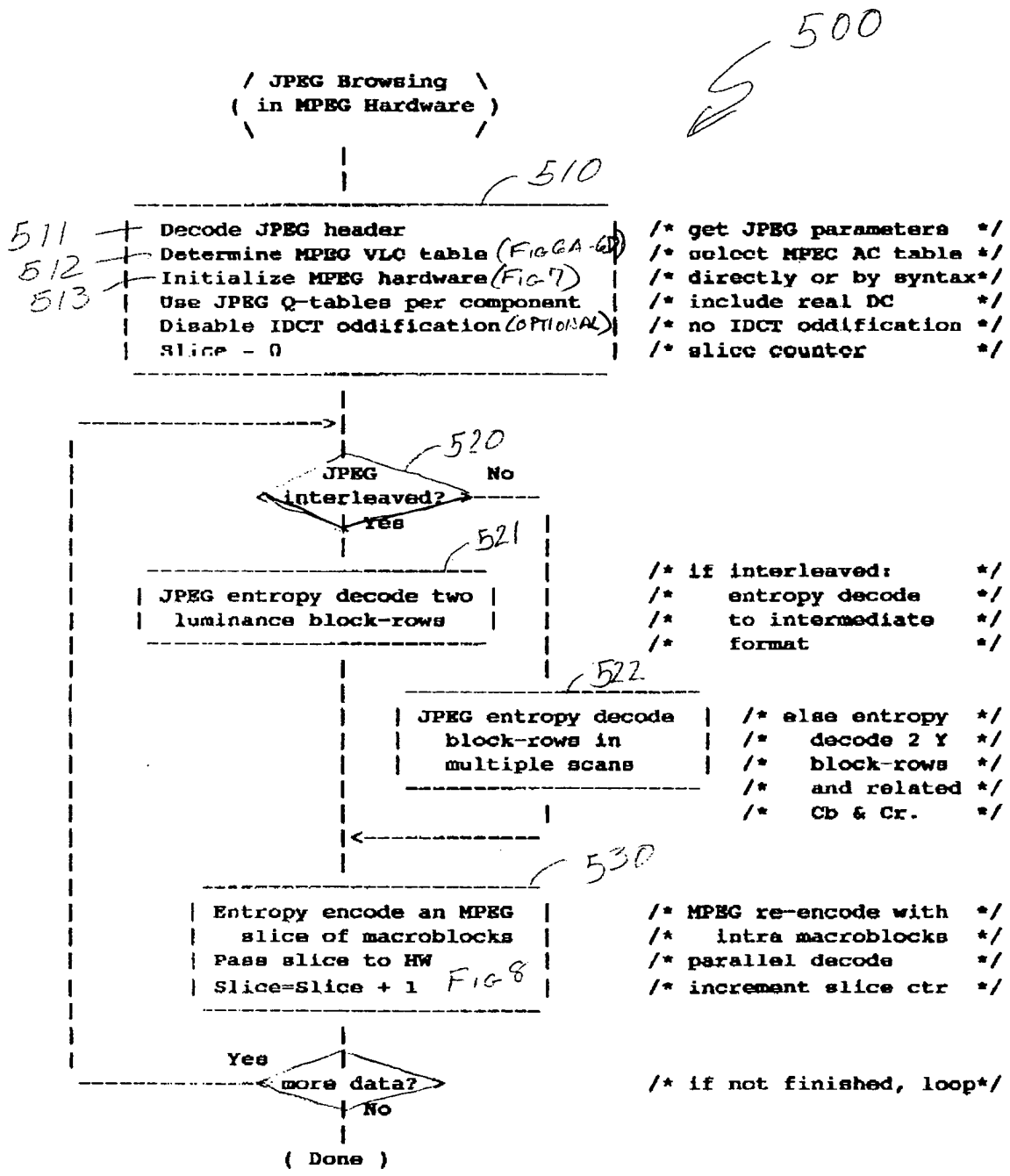
500
JPEG Browsing in MPEG Hardware
510
511
Decode JPEG header
/* get JPEG parameters */
512
Determine MPEG VLC table (Fig 6A-6D)
/* select MPEC AC table */
513
Initialize MPEG hardware (Fig 7)
/* directly or by syntax*/
Use JPEG Q-tables per component
/* include real DC */
Disable IDCT oddification (OPTIONAL)
/* no IDCT oddification */
Slice = 0
/* slice counter */
520
JPEG interleaved?
No
Yes
521
JPEG entropy decode two luminance block-rows
/* if interleaved: */
/* entropy decode */
/* to intermediate */
/* format */
522
JPEG entropy decode block-rows in multiple scans
/* else entropy */
/* decode 2 Y */
/* block-rows */
/* and related */
/* Cb & Cr. */
530
Entropy encode an MPEG slice of macroblocks
/* MPEG re-encode with */
/* intra macroblocks */
Pass slice to HW
/* parallel decode */
Slice=Slice + 1
/* increment slice ctr */
Fig 8
Yes
more data?
/* if not finished, loop*/
No
Done

FIG 5

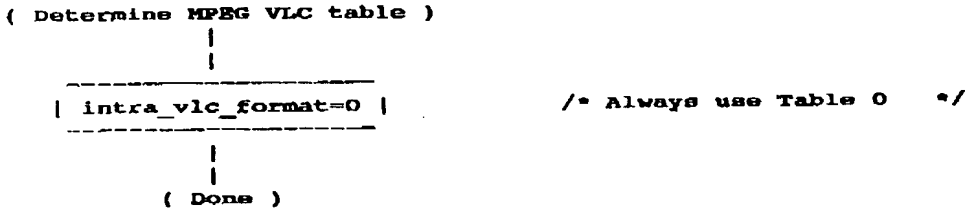
(Determine MPEG VLC table)
| intra_vlc_format=0 |
/* Always use Table 0 */
(Done)

FIG 6A

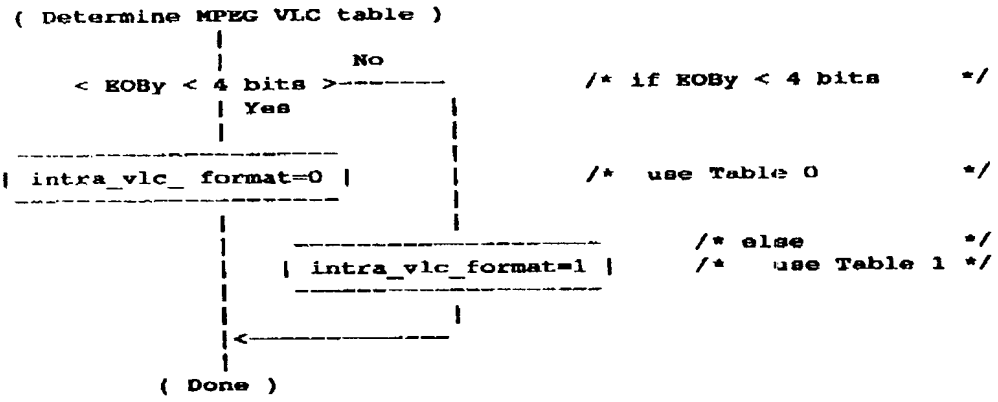
(Determine MPEG VLC table)
No
< EOBy < 4 bits >
Yes
/* if EOBy < 4 bits */
| intra_vlc_ format=0 |
/* use Table 0 */
| intra_vlc_format=1 |
/* else */
/* use Table 1 */
(Done)

FIG 6B

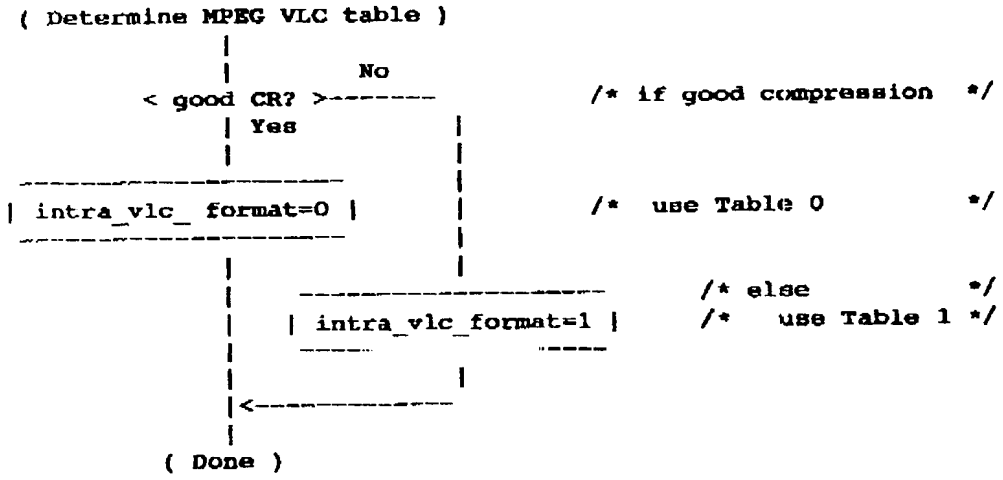
(Determine MPEG VLC table)
No
< good CR? >
Yes
/* if good compression */
| intra_vlc_ format=0 |
/* use Table 0 */
/* else */
| intra_vlc_format=1 |
/* use Table 1 */
(Done)

FIG 6C

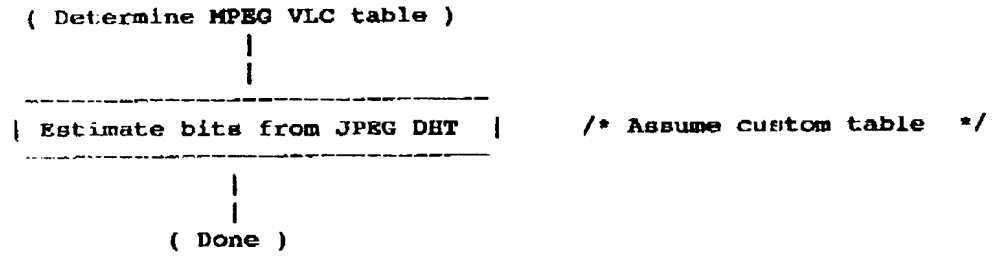
(Determine MPEG VLC table)
| Estimate bits from JPEG DHT |
/* Assume custom table */
(Done)

FIG 6D

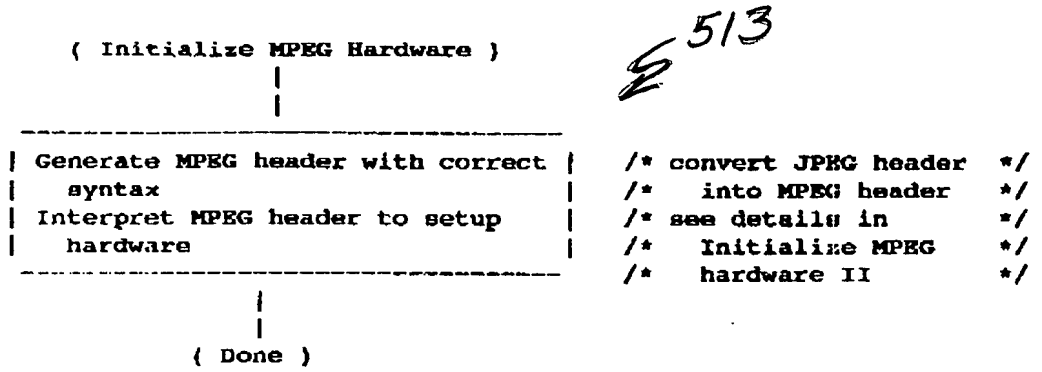
(Initialize MPEG Hardware)
513
Generate MPEG header with correct syntax
Interpret MPEG header to setup hardware
/* convert JPEG header into MPEG header */
/* see details in Initialize MPEG hardware II */
(Done)

FIG 7A

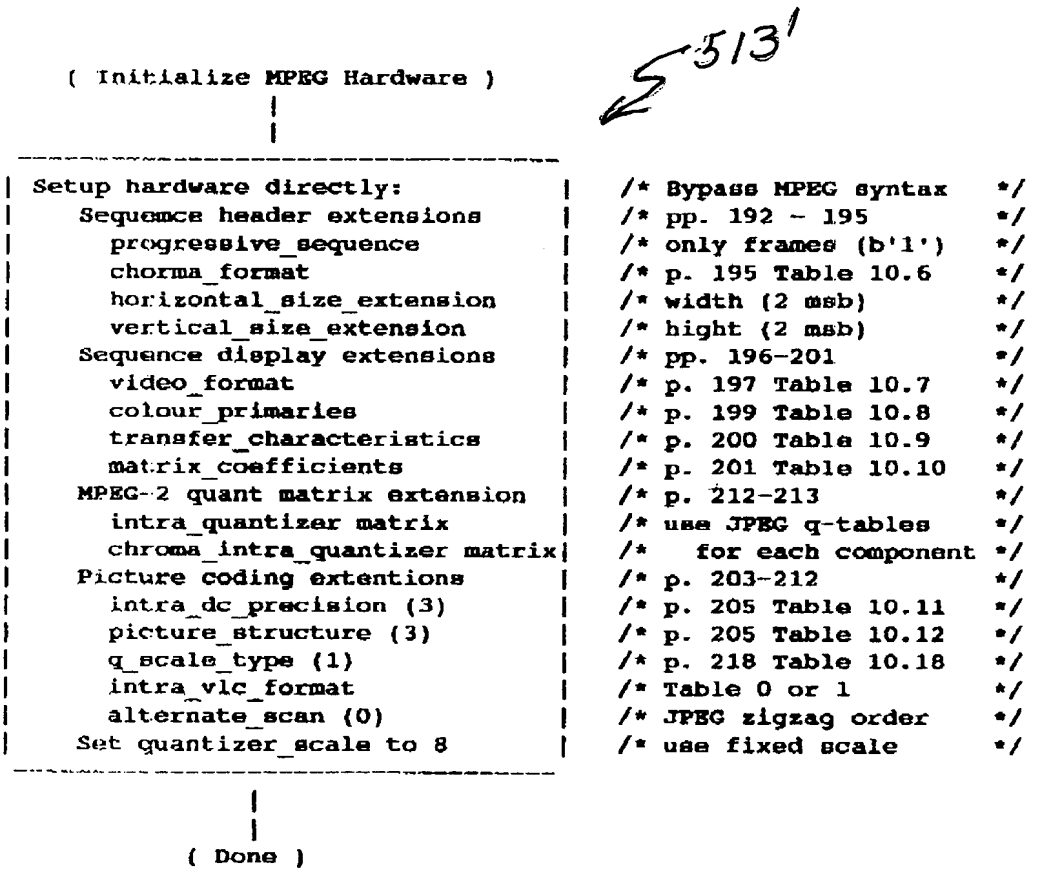
(Initialize MPEG Hardware)
513'
Setup hardware directly:
Sequence header extensions
progressive_sequence
chorma_format
horizontal_size_extension
vertical_size_extension
Sequence display extensions
video_format
colour_primaries
transfer_characteristics
matrix_coefficients
MPEG-2 quant matrix extension
intra_quantizer matrix
chroma_intra_quantizer matrix
Picture coding extentions
intra_dc_precision (3)
picture_structure (3)
q_scale_type (1)
intra_vlc_format
alternate_scan (0)
Set quantizer_scale to 8
/* Bypass MPEG syntax */
/* pp. 192 - 195 */
/* only frames (b'1') */
/* p. 195 Table 10.6 */
/* width (2 msb) */
/* hight (2 msb) */
/* pp. 196-201 */
/* p. 197 Table 10.7 */
/* p. 199 Table 10.8 */
/* p. 200 Table 10.9 */
/* p. 201 Table 10.10 */
/* p. 212-213 */
/* use JPEG q-tables */
/* for each component */
/* p. 203-212 */
/* p. 205 Table 10.11 */
/* p. 205 Table 10.12 */
/* p. 218 Table 10.18 */
/* Table 0 or 1 */
/* JPEG zigzag order */
/* use fixed scale */
(Done)

FIG 7B

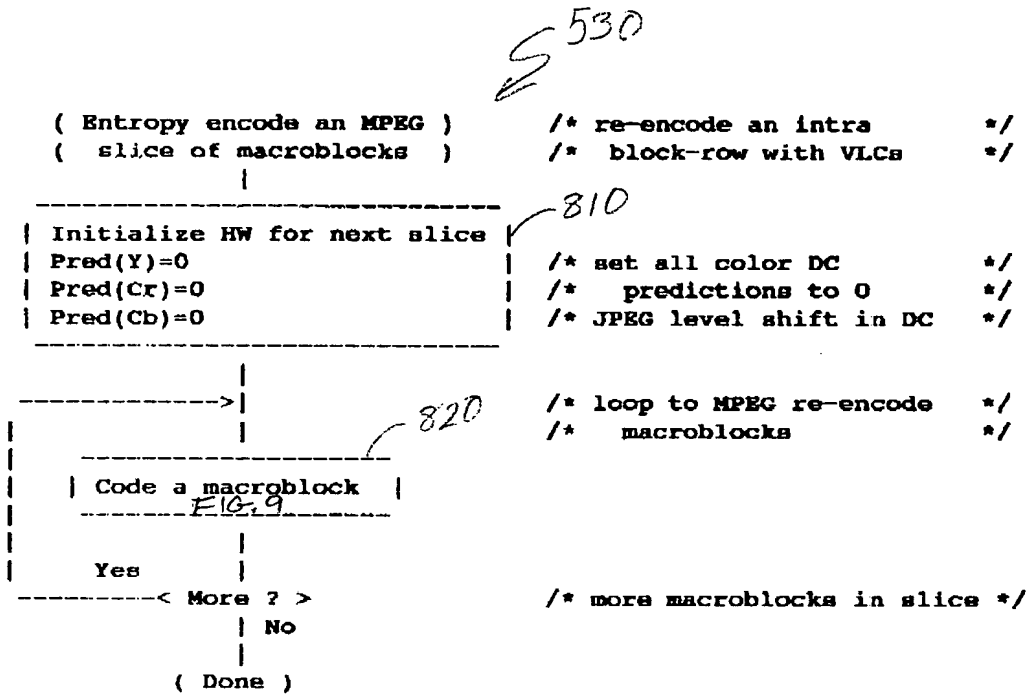
530
(Entropy encode an MPEG) /* re-encode an intra */
(slice of macroblocks) /* block-row with VLCs */
810
Initialize HW for next slice
Pred(Y)=0 /* set all color DC */
Pred(Cr)=0 /* predictions to 0 */
Pred(Cb)=0 /* JPEG level shift in DC */
/* loop to MPEG re-encode */
/* macroblocks */
820
Code a macroblock
FIG. 9
Yes
More ? /* more macroblocks in slice */
No
(Done)

FIG 8

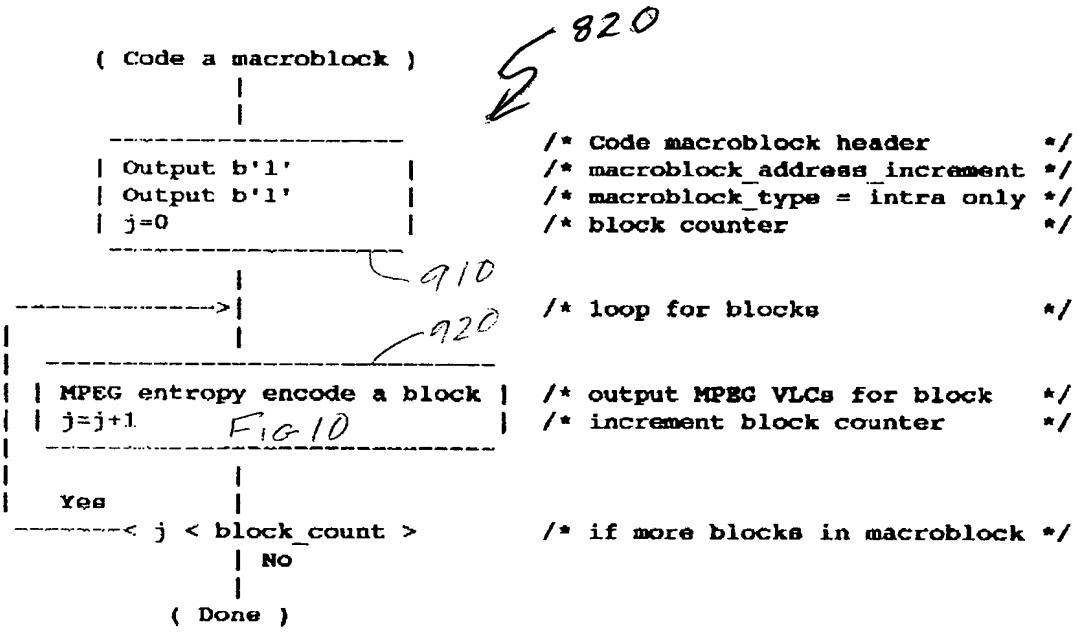
820
(Code a macroblock)
Output b'1' /* Code macroblock header */
Output b'1' /* macroblock_address_increment */
j=0 /* macroblock_type = intra only */
/* block counter */
910
/* loop for blocks */
920
MPEG entropy encode a block /* output MPEG VLCs for block */
j=j+1 /* increment block counter */
FIG 10
Yes
j < block_count /* if more blocks in macroblock */
No
(Done)

FIG 9

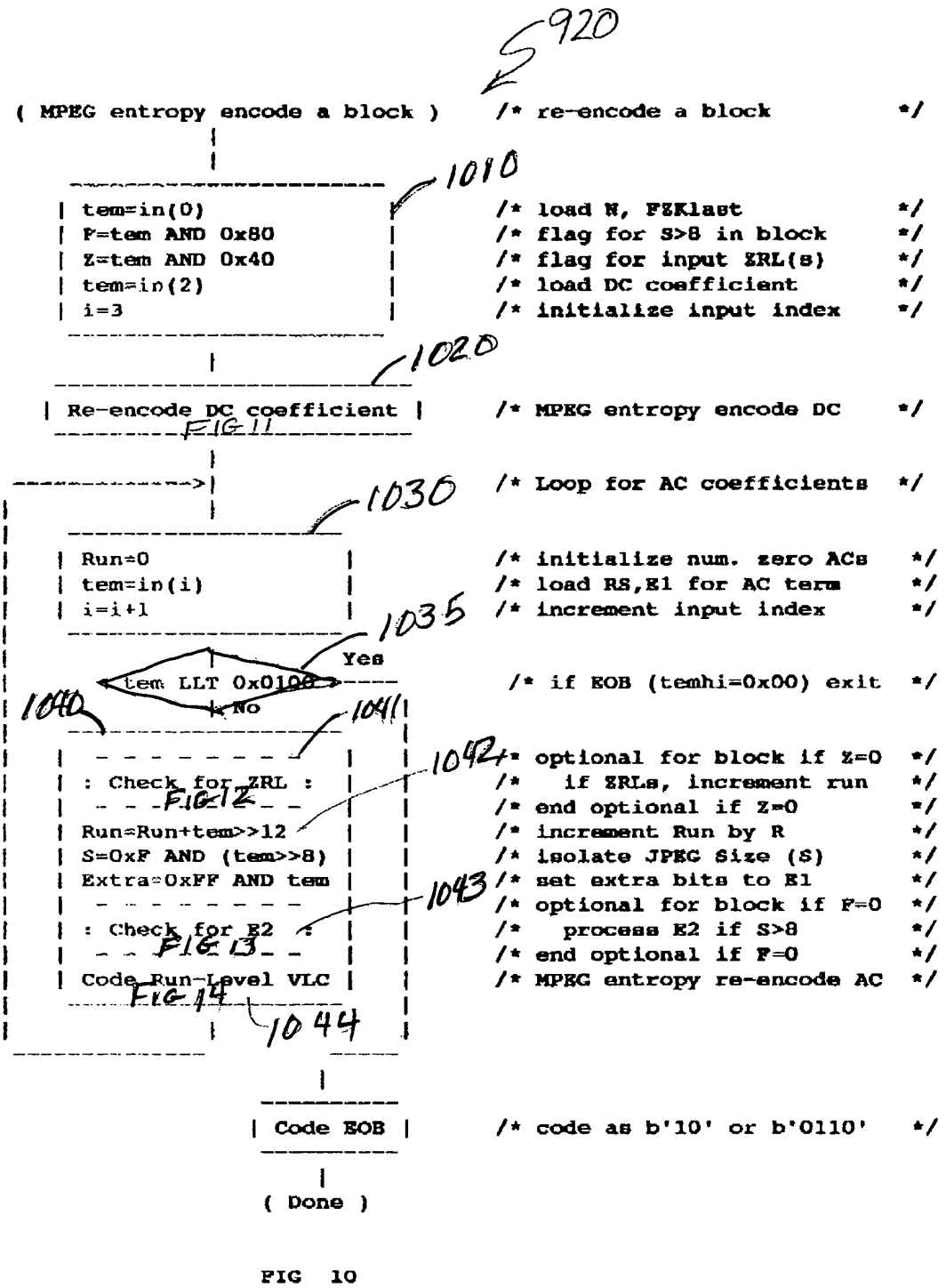
920
(MPEG entropy encode a block)
/* re-encode a block */
1010
tem=in(0)
F=tem AND 0x80
Z=tem AND 0x40
tem=in(2)
i=3
/* load N, FZKlast */
/* flag for S>8 in block */
/* flag for input ZRL(s) */
/* load DC coefficient */
/* initialize input index */
1020
Re-encode DC coefficient
FIG 11
/* MPEG entropy encode DC */
/* Loop for AC coefficients */
1030
Run=0
tem=in(i)
i=i+1
/* initialize num. zero ACs */
/* load RS,E1 for AC term */
/* increment input index */
1035
tem LLT 0x0100
Yes
No
/* if EOB (temhi=0x00) exit */
1040
1041
: Check for ZRL :
FIG 12
Run=Run+tem>>12
S=0xF AND (tem>>8)
Extra=0xFF AND tem
: Check for E2 :
FIG 13
Code Run-Level VLC
FIG 14
1042
/* optional for block if Z=0 */
/* if ZRLs, increment run */
/* end optional if Z=0 */
/* increment Run by R */
/* isolate JPEG Size (S) */
/* set extra bits to E1 */
1043
/* optional for block if F=0 */
/* process E2 if S>8 */
/* end optional if F=0 */
/* MPEG entropy re-encode AC */
1044
Code EOB
/* code as b'10' or b'0110' */
(Done)

FIG 10

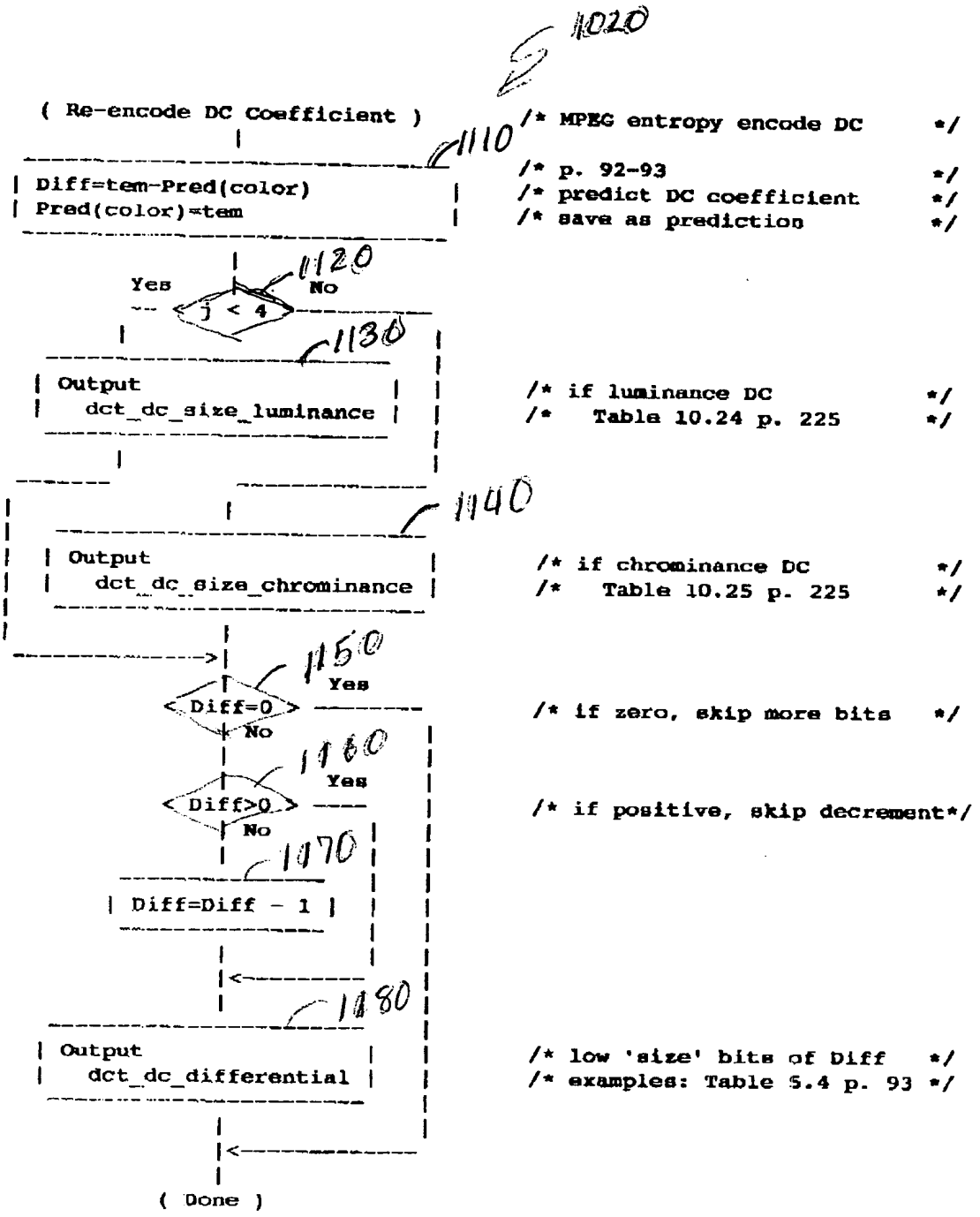
1020
(Re-encode DC Coefficient)
/* MPEG entropy encode DC */
1110
Diff=tem-Pred(color)
Pred(color)=tem
/* p. 92-93 */
/* predict DC coefficient */
/* save as prediction */
1120
Yes
No
j < 4
1130
Output
dct_dc_size_luminance
/* if luminance DC */
/* Table 10.24 p. 225 */
1140
Output
dct_dc_size_chrominance
/* if chrominance DC */
/* Table 10.25 p. 225 */
1150
Yes
Diff=0
No
/* if zero, skip more bits */
1160
Yes
Diff>0
No
/* if positive, skip decrement*/
1170
Diff=Diff - 1
1180
Output
dct_dc_differential
/* low 'size' bits of Diff */
/* examples: Table 5.4 p. 93 */
(Done)

FIG 11

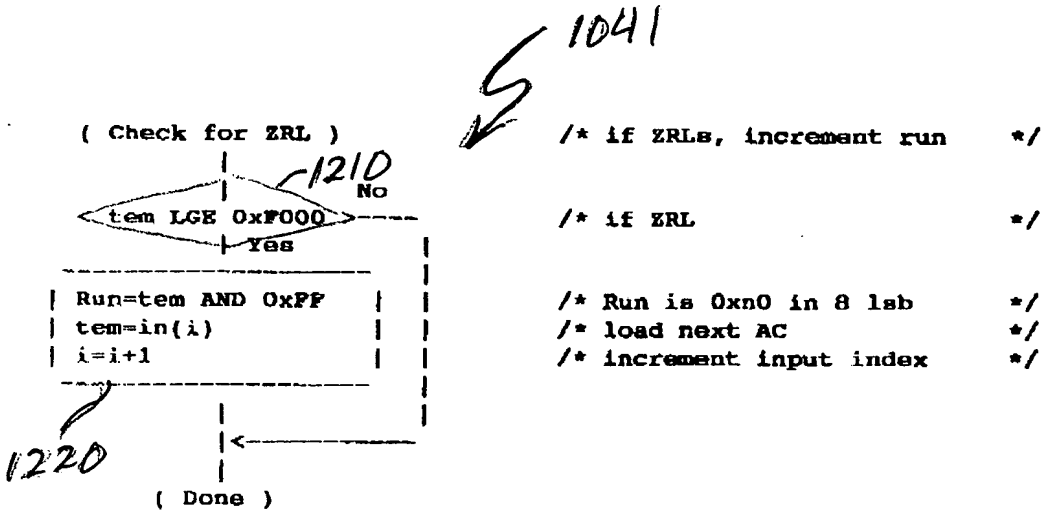
1041
(Check for ZRL)
/* if ZRLs, increment run */
1210
No
tem LGE 0xF000
/* if ZRL */
Yes
Run=tem AND 0xFF
tem=in(i)
i=i+1
/* Run is 0xn0 in 8 lsb */
/* load next AC */
/* increment input index */
1220
(Done)

FIG 12

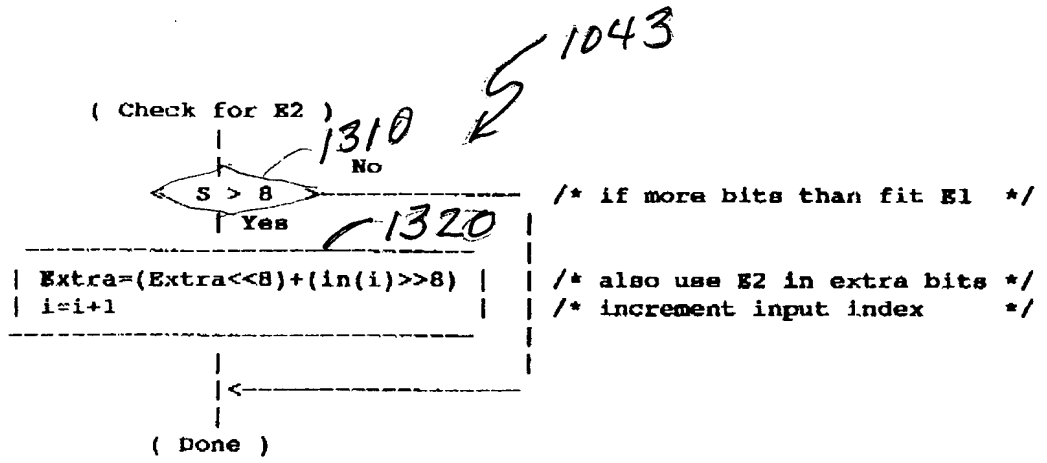
1043
(Check for E2)
1310
No
S > 8
/* if more bits than fit E1 */
Yes
1320
Extra=(Extra<<8)+(in(i)>>8)
i=i+1
/* also use E2 in extra bits */
/* increment input index */
(Done)

FIG 13

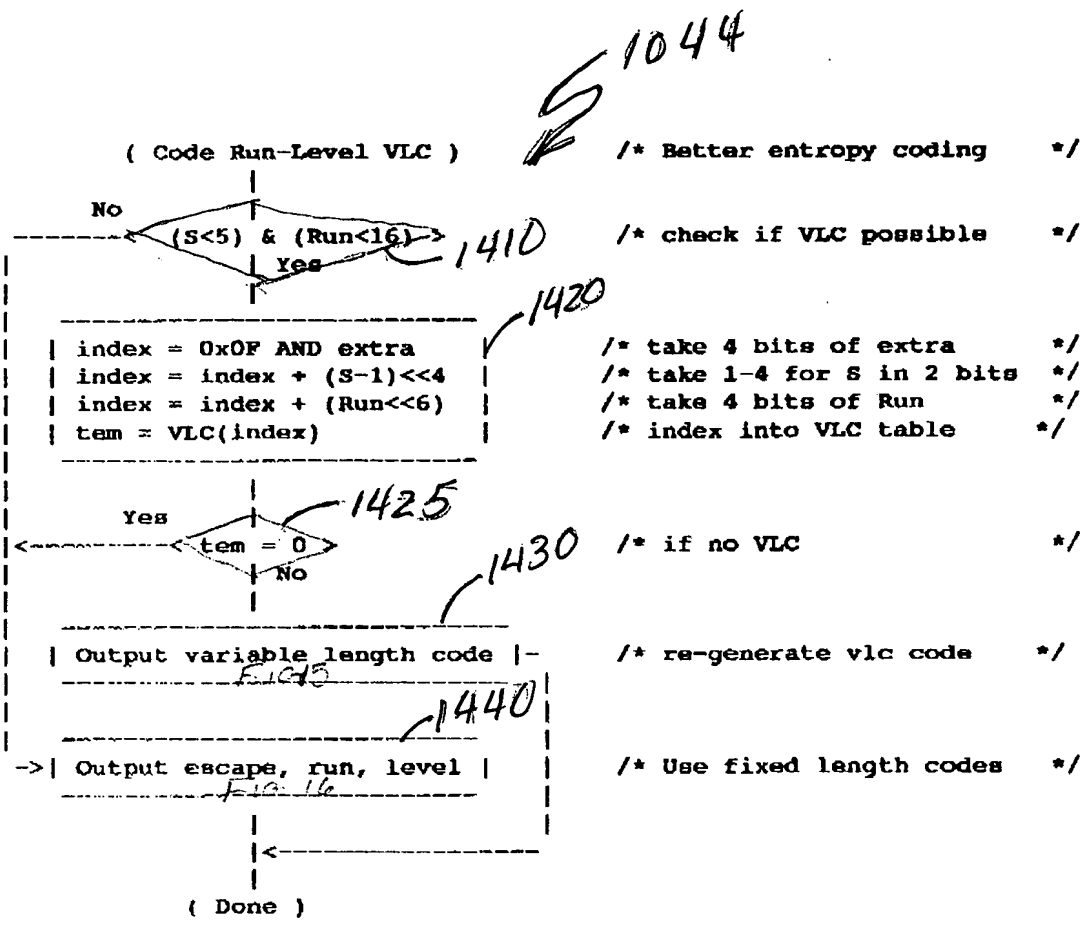

FIG 14A

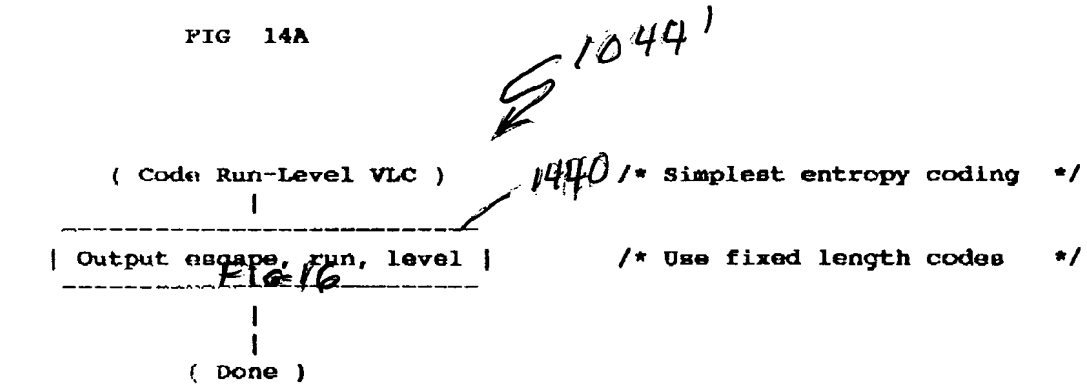

FIG 14B

1430

```
( Output variabl  length code )      /* re-generate vlc code     */
              |                      /*   from Table 1           */
              |                      /* N is byte in VLC         */
              |                      /*   .... .... .... ...0    */
 ---------------------------         /*   0000 000b bbbb bbbb    */
| N=tem                     |        /* save number of bits       */
| tem = table1(index)       |        /* look up code              */
| Output N bits of tem      |        /* output 2-17 lsb bits      */
 ---------------------------
              |
              |
         ( Done )
```

FIG 15A

1430

```
( Output variable length code )      /* re-generate vlc code     */
              |                      /*   from Table 4           */
              |                      /*   00bb bbbb bbbn nnnn    */
 ---------------------------
| N=0x1F AND tem            |        /* calculate number of bits*/
| tem = tem >> 5            |        /* shift bits to lsb        */
| Output N bits of tem      |        /* output 2-17 lsb bits     */
 ---------------------------
              |
              |
         ( Done )
```

FIG 15B

1430

```
( Output variable length code )      /* re-generate vlc code     */
              |                      /*   from Table 5           */
              |                      /* VLC contained zzzznnnn   */
 ---------------------------
| N= tem >> 4               |        /* number of leading zeros */
| Output N zeros            |        /* output leading zeros    */
| Output '1'                |        /* output one              */
| N=0x0F AND tem            |        /* calculate remaining bits*/
| tem=table5(index)         |        /* get byte from Table 5   */
| Output N bits of tem      |        /* output remaining bits   */
 ---------------------------
              |
              |
         ( Done )
```

FIG 15C

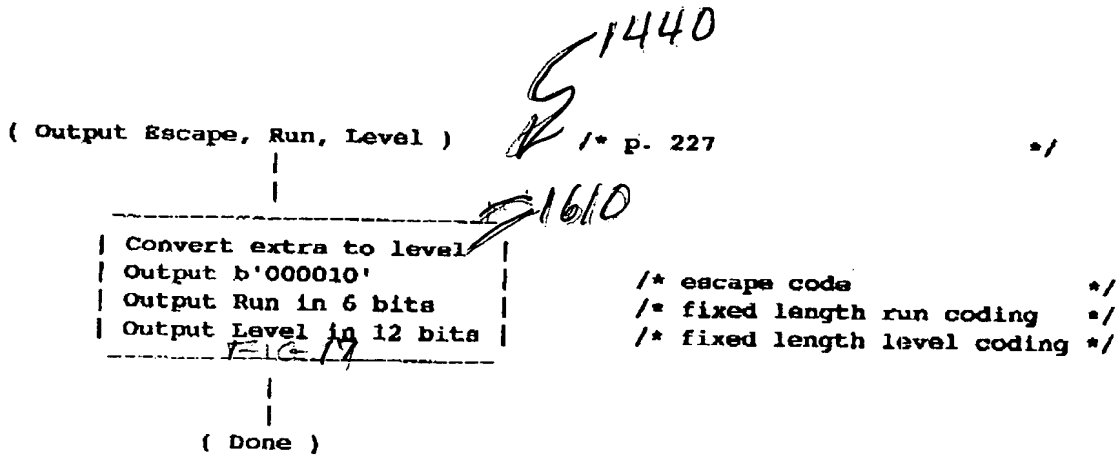
1440
(Output Escape, Run, Level)
/* p. 227 */
1610
Convert extra to level
Output b'000010'
Output Run in 6 bits
Output Level in 12 bits
/* escape code */
/* fixed length run coding */
/* fixed length level coding */
FIG 17
(Done)

FIG 16

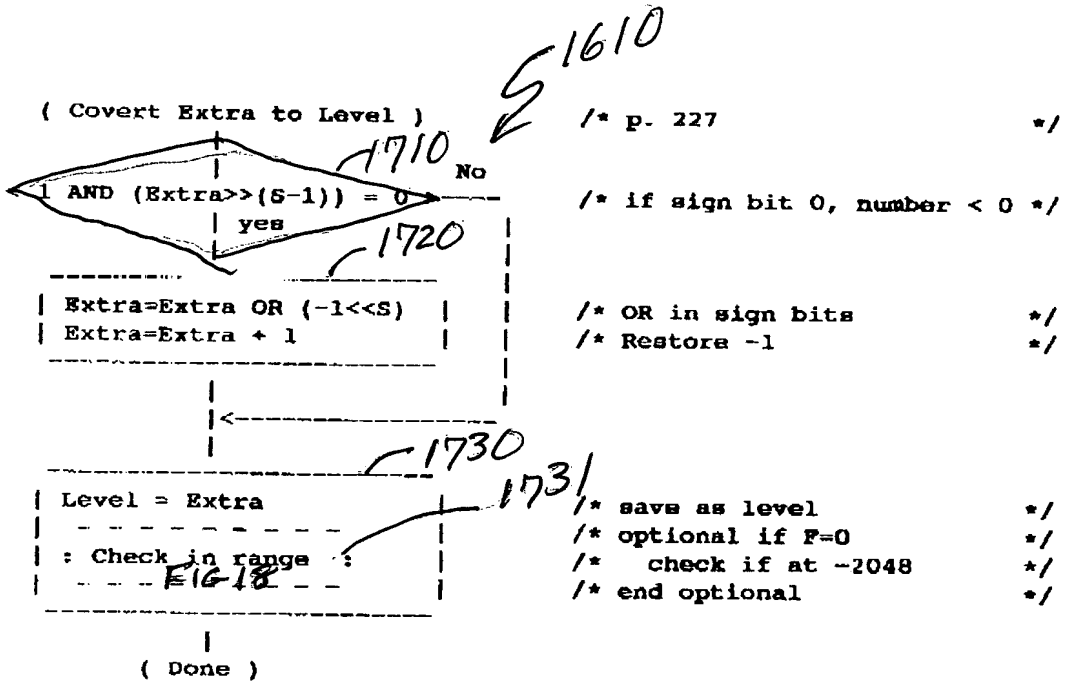
1610
(Covert Extra to Level)
/* p. 227 */
1710
No
1 AND (Extra>>(S-1)) = 0
yes
/* if sign bit 0, number < 0 */
1720
Extra=Extra OR (-1<<S)
Extra=Extra + 1
/* OR in sign bits */
/* Restore -1 */
1730
1731
Level = Extra
: Check in range :
FIG 18
/* save as level */
/* optional if F=0 */
/* check if at -2048 */
/* end optional */
(Done)

FIG 17

1731

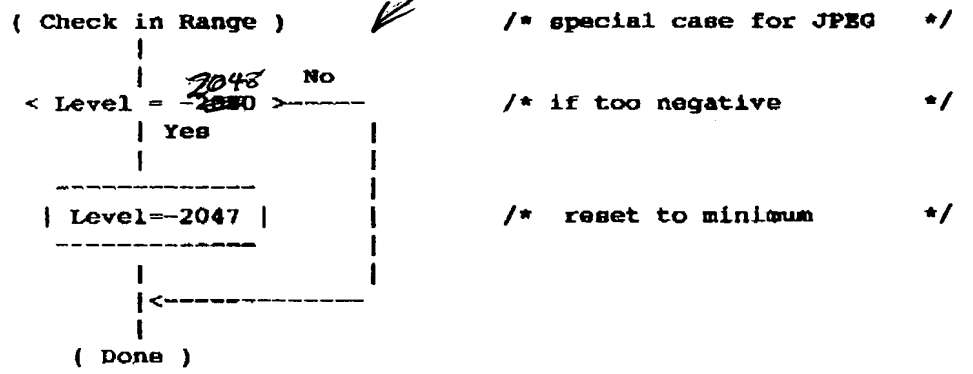

FIG 18A

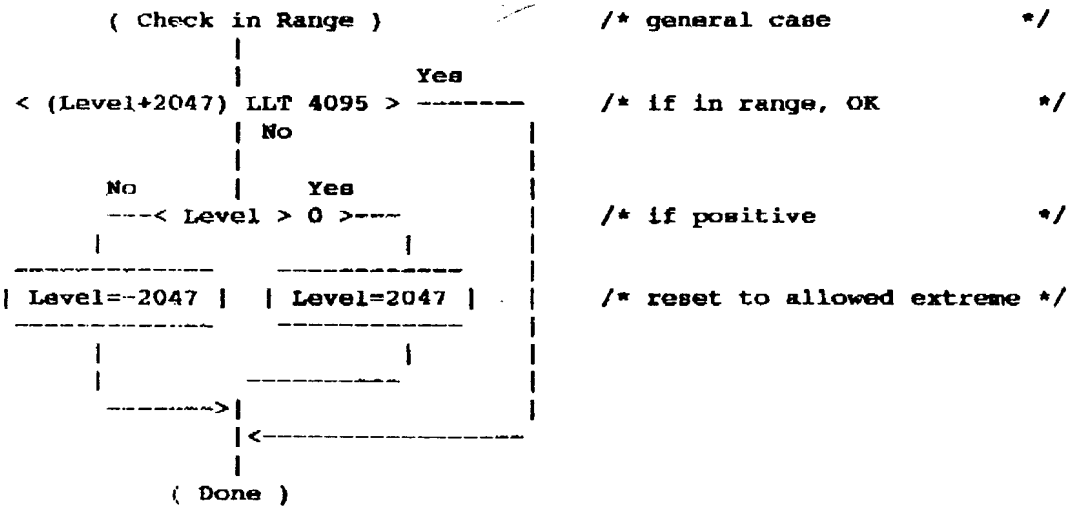

FIG 18B